Received: 10 March 2017 | Accepted: 23 July 2017
DOI: 10.1002/cae.21867

RESEARCH ARTICLE

WILEY

Optimization of chemical engineering problems with EMSO software

João Paulo Henrique[1] | **Ruy de Sousa Jr.**[1,2] | **Argimiro R. Secchi**[3] | **Mauro A. S. S. Ravagnani**[4] | **Caliane B. B. Costa**[4]

[1] Chemical Engineering Graduate Program, Universidade Federal de São Carlos (UFSCar), São Carlos, São Paulo, Brazil

[2] Chemical Engineering Department, Universidade Federal de São Carlos (UFSCar), São Carlos, São Paulo, Brazil

[3] Chemical Engineering Program, COPPE, Universidade Federal do Rio de Janeiro (UFRJ), Cidade Universitária, Rio de Janeiro, Rio de Janeiro, Brazil

[4] Chemical Engineering Department, Universidade Estadual de Maringá, Maringá, Paraná, Brazil

Correspondence
Caliane B. B. Costa, Chemical Engineering Department, Universidade Estadual de Maringá, Av. Colombo, 5790 Bloco D90, CEP 870208290, Maringá, Paraná, Brazil.
Email: cbbcosta@uem.br

Funding information
CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior); CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), Grant number: 132162/2015-6; FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), Grant number: 2011/51902-9

Abstract

EMSO software, a free tool for teaching and academic research, presents a favorable environment for simulation and optimization of chemical engineering problems. Although its use in research activities have been demonstrated in the literature, its application as a supporting tool for educational purposes in process systems engineering courses has not been evaluated. The objective of this paper is to demonstrate that this software is suitable as a tool for educational purposes in process systems engineering courses at undergraduate (complementary elective subject) and graduate levels. To accomplish this task, different cases studies, encompassing different sort of programming formulations, are proposed and solved with different solvers in EMSO.

KEYWORDS
educational tool, EMSO software, MILP, MINLP, NLP

1 | INTRODUCTION

Currently, to be competitive, companies must plan and manage their production line in order to maximize profits and decrease waste. To make this possible in design or during operation stage, engineers, and designers may use optimization methods and software available in the market. The tasks involved in this activity are part of the process systems engineering (PSE) area, which uses computer-aided approaches in order to predict the performance of a chemical process, optimize or even control it. The tools used by PSE professionals must be suitable to deal with the nature of these processes, which are often highly integrated [7].

Several chemical process simulators are available, which can be divided into two groups: sequential modular simulators, such as Aspen Plus®, Aspen HYSYS®, CHEMCAD®, and ProSim®; and equation-oriented simulators, such as gPROMS®, Aspen Dynamics®, and EMSO (Environment for Modeling, Simulation, and Optimization) [25].

Many researchers have used commercial sequential modular simulators for simulation, optimization, and integration with other programs [3,9,22]. However, these commercial tools do not offer the possibility of inspection and edition of equipment models by the users.

Software EMSO has the advantage that all developed models are open for inspection and extension by any user, which makes it extremely flexible for research use and is free of charge for academic research and educational activities [23,25]. Furthermore, its applicability is wide, providing tools for steady and dynamic simulations, parameter

estimation, data reconciliation, sentitivity analysis, and optimization. For instance, Cunha et al. [5] developed a dynamic model for compressors in a Natural Gas Liquid Recovery Unit. Furlan et al. [11] and Costa et al. [4] studied a sugarcane biorefinery of first and second generation and proposed the solution of different optimization problems to improve the production of bioelectricity and bioethanol simultaneously with the decrease in liquid effluent (concentrated vinasse) output and CO_2 emissions. Other researchers developed plugins that improve results, facilitate convergence of the optimization codes [12,27], and make possible the communication between EMSO and other software [18].

EMSO features and its successful application in research PSE activities pave the way for the evaluation of this software as a supporting tool in PSE courses at undergraduate (complementary elective subject) and graduate levels. Therefore, this paper aims to demonstrate the versatility of this software in solving optimization problems with different levels of complexity, which makes it suitable as a tool for educational purposes in PSE courses. To accomplish this objective, the main features of EMSO are presented, including explanations for new users and of the solvers suitable for the different sorts of optimization problems. Then, different optimization problems in the field of process systems engineering, with increasing levels of complexity, are proposed. These case studies include design of continuous stirred tank reactors, production planning, synthesis of heat exchanger network, and definition of operating conditions in a sugarcane biorefinery. They cover a wide range of problems: NLP (Nonlinear Programming), MILP (Mixed-Integer Linear Programming), MINLP (Mixed-Integer Nonlinear Programming) and NLP in a MOO (Multiobjective Optimization). Finally, the proposed case studies are solved in EMSO and aspects related to them, such as required computational time and number of variables, are discussed.

2 | USING EMSO SOFTWARE FOR EDUCATIONAL PURPOSES

Developed by Soares and Secchi [25], EMSO has a graphical interface for the construction of models based on block diagram, and a modeling language based on the concept of object-oriented approach that allows the addition of parameters, variables, and equations to hierarchically increase the model complexity. For beginners, the user manual provides the basic instructions for using the available examples and how the user can build her/his models. Despite being based on C++ language, it is not necessary a previous knowledge on this programming language, since EMSO has its own modeling language, which is simple and uses the concepts of inheritance and composition. EMSO Reference Manual, available in the installation folder of the software, describes how to develop a new plugin (solver) using the most common scientific programming languages (C, C++, and Fortran) and how to implement it into the software [24].

Figure 1 shows EMSO graphical user interface (GUI) with an example model named *NLP_cstr.mso*. Besides the common menu bar (label 1 in the figure), the interface exhibits the model (label 2), which can be one of those already present in EMSO model library (*eml*, label 3). All models in EMSO are open for inspection and edition. In addition, the user can create a new model from the beginning. Data related to model, such as the number of variables, equations, specifications, and degrees of freedom, are exhibited in the Console (label 4 in Figure 1). In Results panel (label 5), user can check the results after running the model. For more details about EMSO environment software, the reader is referred to Rodrigues et al. [23]. To solve an optimization problem, the user must use a consistent model, that is, a model with zero degree of freedom, and indicate which of the model specifications should be set free, as degrees of freedom. This task is demonstrated further is this paper, in section 3.1, in the first case study presentation.

Several differential-algebraic equations (DAE), nonlinear algebraic equations (NLA), and optimization solvers are already available in EMSO. However, other solvers were also added to complete the tools applied to solve the proposed simulations and optimization tasks of this work. In the remainder of this section, the solvers used for solving the case studies proposed in this work are described briefly.

2.1 | Nonlinear programming (NLP) solver

To solve nonlinear programming problems, the IPOPT (Interior Point OPTimizer) algorithm, released by Wächter and Biegler [30] and already implemented in the software original solver library, can be used. It is suitable for large-scale linear and nonlinear optimization problems with up to millions of variables and constraints. The general nonlinear programming problem is:

$$\begin{aligned} & \min_{\mathbf{x}} \ f(\mathbf{x}) \\ s.t. \quad & \\ & g_j^L \le g_j(\mathbf{x}) \le g_j^U, \quad j = 1, 2, \ldots, m \\ & \mathbf{x}^L \le \mathbf{x} \le \mathbf{x}^L \\ & \mathbf{x} \in X \subseteq \Re^n \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the optimization variables vector with lower $\mathbf{x}^L \in \Re^n$ and upper $\mathbf{x}^U \in \Re^n$ bounds, $f(\mathbf{x}) : \Re^n \to \Re$ is the objective function, and $g_j(\mathbf{x}) : \Re^n \to \Re^m$ are the constraint functions. The g_j^L and g_j^U are specified lower and upper bounds on the constraint functions with $g_j^L \le g_j^U$. If $g_j^L = g_j^U$, the j-th constraint is an equality constraint.

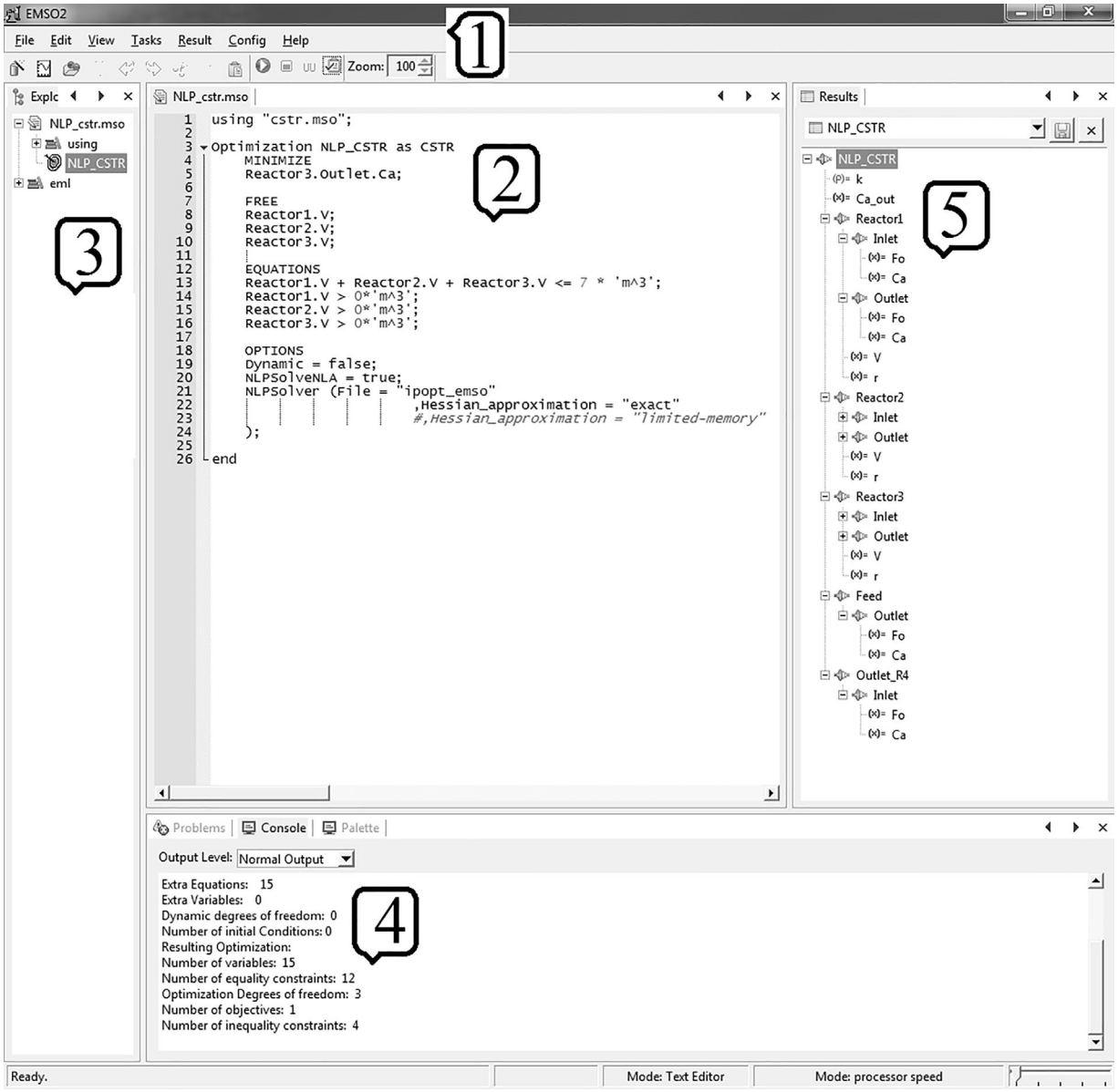

FIGURE 1 EMSO graphical user interface

According to Wächter [28], functions $f(\mathbf{x})$ and $g(\mathbf{x})$ can be linear, nonlinear, convex or nonconvex, but should be sufficiently smooth and continuously differentiable at least once. It is important to keep in mind that the algorithm is only trying to find a local minimum of the problem and, possibly, second derivatives might not be available. Considering it, if the Hessian approximation is defined as "exact" in the software, the algorithm computes first and second derivatives of the problem functions $f(\mathbf{x})$ and $g(\mathbf{x})$, and it normally converges in fewer iterations and is more robust than if "limited-memory" option is used. It is possible to approximate Hessian matrix by means of a limited-memory quasi-Newton method in problems for which the second derivative might not be available [28]. More details of the algorithm can be found in the literature [28–30].

2.2 | Deterministic mixed-integer nonlinear programming (MINLP) solver

The deterministic algorithm BONMIN *(Basic Open-source Nonlinear Mixed INteger programming)* is available as a solver in EMSO library. It can be used to solve general MINLP problems, formulated as [2]:

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}) \\ s.t. \\ &h(\mathbf{x},\mathbf{y}) = 0 \\ &g(\mathbf{x},\mathbf{y}) \le 0 \\ &\mathbf{x} \in X \subseteq \Re^n \\ &\mathbf{y} \in Y \subseteq Z^m \end{aligned} \tag{2}$$

where $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ is a vector of n continuous variables, $\mathbf{y} = [y_1, y_2, ..., y_m]^T$ is a vector of m integer variables, $h(\mathbf{x}, \mathbf{y}) = 0$ are p equality constraints, $g(\mathbf{x}, \mathbf{y}) \leq 0$ are q inequality constraints, and $f(\mathbf{x}, \mathbf{y})$ is the objective function [10].

Configuring the options of BONMIN, it is possible to choose among Branch-and-Bound (B-BB) [14], Outer-Approximation decomposition (B-OA) [8], Quesada and Grossmann's branch-and-cut (B-QG) [20], and Hybrid outer-approximation based branch-and-cut (B-Hyb) [2] algorithms variations.

2.3 | Non-deterministic mixed-integer nonlinear programming (MINLP-PSO) solver

Particle Swarm Optimization method was proposed by Kennedy and Eberhart [15] and is based on the patterns of nature, like the movement of flock of birds or shoals of fish. In this non-deterministic algorithm, each particle in the swarm moves in the search space looking for the optimal position based on its own movement history and on the entire swarm movement history. For more details about PSO method, the reader is referred to the literature [15–17].

A code for solving MINLP by the non-deterministic Particle Swarm Optimization method was implemented and added to the software library. It was named MINLP-PSO and was developed using feasible path approach (equality constraints must be satisfied) [1]. The implemented algorithm solves the MINLP problems by relaxing the integer variables, and, at the end of each iteration, each relaxed integer variable is rounded to the closest integer value. The code was developed considering as default for the cognitive (c_1) and social (c_2) parameters and for the inertia weight factor (w) the values of 2.0, 2.0, and 1.0, respectively.

2.4 | Multiobjective optimization (MOO-PSO) solver

The multiobjective PSO code was developed by Gonçales et al. [13] and was coupled to EMSO as a solver (MOO-PSO), using the feasible path approach. Besides that, it adopts the non-dominance concept when comparing solutions [13,21], and generates a repository containing all non-dominated particles found on each iteration. This code incorporates the mechanism of crowding distance (CD) in order to maintain non-dominated particles better distributed in the repository, deleting some crowded particles. Particles are arranged in CD order and global best is selected among non-dominated particles located in least populated regions. Since non-dominated particles are identified, it is possible to plot the Pareto front, which helps the decision-makers to choose which one of the non-dominated solutions better suits their goals [13,21].

The previous default values for cognitive (c_1) and social (c_2) parameters, and the inertia weight factor (w) were also considered here. For more information, the reader is referred to the work of Gonçales et al. [13].

3 | CASE STUDIES

In this paper, different applications of optimization in PSE using EMSO are presented, in increasing level of difficulty, for didactic purpose. As already exposed, they cover a wide range of problems and the approach used to solve them are discussed in their presentation.

3.1 | Continuous stirred tank reactors in series

In this application, a model with three Continuous Stirred Tank Reactors (CSTRs) in series was developed. In these reactors, a reaction that consumes component A takes place. The objective was to determine the volumes of Reactors 1, 2, and 3 that minimize the concentration of reactant at the outlet of the third reactor (C_{A3}). There is a constraint that the total volume (i.e., the sum of the volumes of all three reactors) must be less than or equal to $7.0\,m^3$.

The model was implemented with molar balances and reactant consumption rate in each reactor, represented by Equations 3 and 4, respectively.

$$F_0(C_{Ai-1} - C_{Ai}) = r_{Ai}V_i, i \in \{1, 2, 3\} \quad (3)$$

$$r_{Ai} = k\,C_{Ai}^{1.5}, i \in \{1, 2, 3\} \quad (4)$$

where the reaction constant k is $1.0\,m^{1.5}/(hr\,kmol^{0.5})$, the volumetric flow rate F_0 is $2.0\,m^3/hr$ and the feed concentration of reactant A, C_{A0}, is $1\,kmol/m^3$. Figure 2 presents in detail the development of the model in EMSO software. Initially, for the declaration of parameters and variables, model *stream_model.mso* was developed, as shown in Figure 2a. This model is then used in the reactor model *reactor_model.mso*, shown in Figure 2b. Finally, Figure 2c presents the Flowsheet for the problem of the three CSTRs in series, with zero degrees of freedom. It is important to note that the same reactor model is used three times (lines 13–15 of Figure 2c).

The NLP optimization model of this application is given by Problem 5 and was solved using IPOPT solver, as shown in Figure 3. In order to optimize the system, it was necessary to free the variables in the software (lines 7–10 of Figure 3), because these were specified in the previously presented Flowsheet (lines 28–30 of Figure 2c).

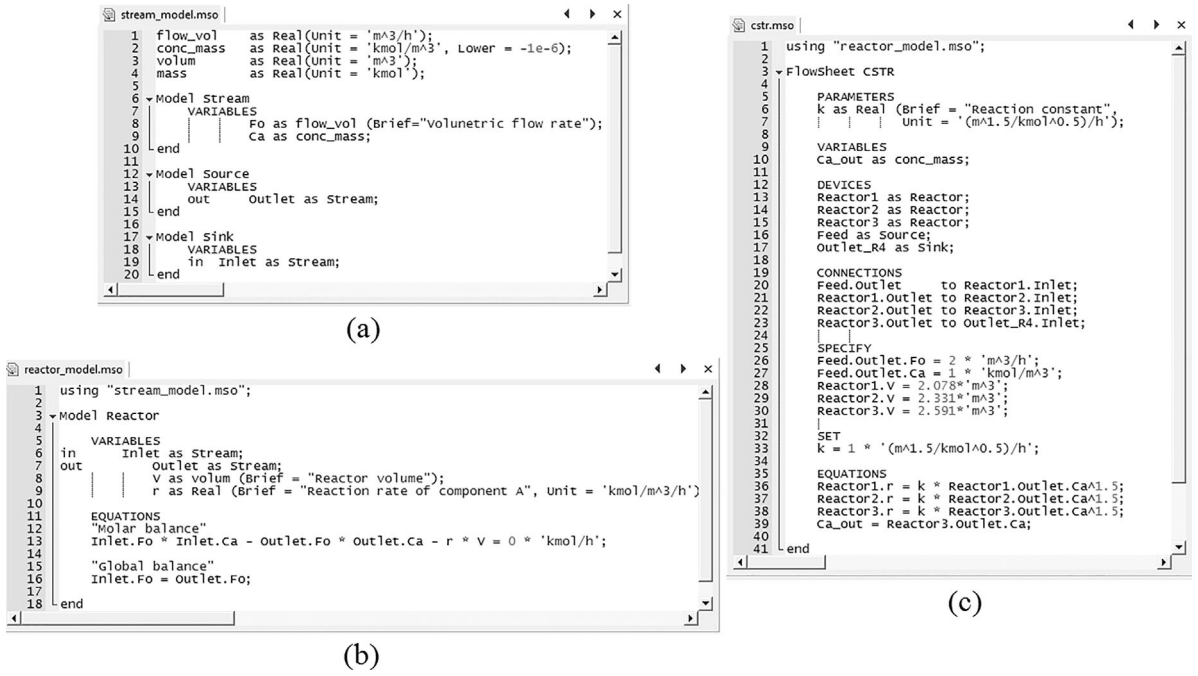

FIGURE 2 Stream, reactor, and flowsheet models for the three CSTRs in series in EMSO software

$$\begin{aligned} &\min_{\mathbf{V}} \; C_{A3}(\mathbf{V}) \\ &s.t. \\ &Eq.(5); Eq.(6); \\ &\sum_{i=1}^{3} V_i - 7.0 \leq 0 \\ &\mathbf{V} = [V_1, V_2, V_3]^T \\ &\mathbf{V} \in \Re_+^3 \end{aligned} \tag{5}$$

To verify the convergence rate to the best solution, the problem was solved twice, defining the Hessian approximation both as exact and limited-memory.

FIGURE 3 Optimization of the three CSTRs in series in EMSO software

3.2 | Production planning

This problem was described by Biegler et al. [1] and the objective was to maximize the profit of a supplier of chemical component C. This component can be produced using reactant B, either with Process II or III. On the other hand, reactant B can either be bought from another supplier or be produced with Process I, using component A as raw material. The maximum supply of A is 16 tonne/hr. The purchase price of chemicals A and B are 500.00 and 950.00 $/tonne, respectively. The maximum demand for C is 15 tonne/hr, and the selling price is 1,800.00 $/tonne for a supplying rate up to 10 tonne/hr. Any amount that exceeds this limit is sold at 1,500.00 $/tonne. Table 1 presents the data related to this application.

In the approach adopted to solve this problem, binary variables $\boldsymbol{y}$ are decision variables representing whether each process is chosen (e.g., y_{A1} is equal to 1 if raw material A is purchased, and 0 otherwise). On the other hand, variables $\mathbf{x}$ represent mass flow rates in tonne/hr. Figure 4 presents the superstructure for the synthesis of the production route of chemical product C, and Figure 5 presents this model in EMSO software.

After C is produced in Process II (x_{C2}) or III (x_{C3}), there are two possibilities, according to the physical constraints imposed by the problem statement: either the sum ($x_{C2} + x_{C3}$) is less than or equal to 10 tonne/hr or it is between 10 and 15 tonne/hr. In the approach adopted, if the sum is less than or equal to 10 tonne/hr, the binary variable y_{D1} receives value 1 ($y_{D2} = 0$) and the continuous variable (x_{CD1}) receives the value of the sum ($x_{C2} + x_{C3}$). Otherwise, y_{D2} receives value 1 ($y_{D1} = 0$) and the continuous variable (x_{CD2}) receives the value of the sum ($x_{C2} + x_{C3}$). The optimization model has a

TABLE 1 Cost data and conversion processes in the production planning optimization problem

Process	Conversion	Capital and operating costs (*CC* and *OC*)	
		CC ($/tonne)	*OC* ($/tonne raw material)
I	90% A to B	1,000.00	250.00
II	82% B to C	1,500.00	400.00
III	95% B to C	2,000.00	550.00

MILP formulation (Problem 6) and can be developed in EMSO environment as presented in Figure 6. It is important to draw attention to the fact that, to the best of authors' knowledge, there is no previous report in the literature of MILP application using EMSO software.

$$\max_{\mathbf{x},\mathbf{y}} P = 100[18x_{CD1} + 180y_{D2} + 15(x_{CD2} - 10y_{D2}) - 10y_{A1} - 7.5x_{A1} - 9.5x_{B0} - 15y_{B2} - 4x_{B2} - 20y_{B3} - 5.5x_{B3}]$$

$$s.t.$$

$$\begin{array}{lll} x_{B1} - 0.90x_{A1} = 0; & y_{B2} + y_{B3} - 1 = 0; & x_{B2} + x_{B3} - x_{B1} - x_{B0} = 0; \\ x_{C2} - 0.82x_{B2} = 0; & x_{C3} - 0.95x_{B3} = 0; & x_{CD1} + x_{CD2} - x_{C2} - x_{C3} = 0; \\ y_{D1} + y_{D2} - 1 = 0; & x_{A1} - 16y_{A1} \le 0; & x_{B0} - 50y_{B0} \le 0; \\ y_{A1} + y_{B0} - 2 \le 0; & x_{B2} - 30y_{B2} \le 0; & x_{B3} - 30y_{B3} \le 0; \\ x_{C2} - x_{C3} - 15 \le 0; & x_{CD1} - 10y_{D1} \le 0; & x_{CD2} - 15y_{D2} \le 0;\ 10y_{D2} - x_{CD2} \le 0 \end{array} \quad (6)$$

$$\mathbf{x} = [x_{A1}, x_{B0}, x_{B1}, x_{B2}, x_{B3}, x_{C2}, x_{C3}, x_{CD1}, x_{CD2}]^T$$
$$\mathbf{y} = [y_{A1}, y_{B0}, y_{B2}, y_{B3}, y_{D1}, y_{D2}]^T$$
$$\mathbf{y} \in \{0,1\}^6$$
$$\mathbf{x} \in \Re_+^9$$

To solve this optimization problem, BONMIN solver was used.

3.3 | Heat exchanger network

In this MINLP optimization problem, presented by Turkay and Grossmann [26], three heat exchangers operating in countercurrent are connected according to Figure 7. The objective is to operate the network with the lowest total annual cost, considering the cost of utilities in cooler 2 and heater 3 and the capital cost of each device, which are functions of their areas. Data relating to temperature, flow rates and costs are given in Table 2.

It is considered that ($T_{H\_inlet} - T_2$) and ($T_1 - T_{C\_inlet}$) are greater than or equal to 10 K [1].

Problem 7 describes the problem with continuous variables A_j, T_1, T_2, $q_{H\_LN\_2}$, and $q_{C\_LN\_3}$, which represent, respectively, the area of heat exchanger j, outlet temperature of hot stream from heat exchanger 1, outlet temperature of cold stream from heat exchanger 1, heat load from hot stream in cooler 2, and the heat load from cold stream in heater 3, respectively. As the capital cost function changes according to the area of heat exchangers, which can be within three different ranges, this problem can be written as a problem with disjunctions. In the approach adopted, the binary variables y_{ij} represent the choice of an area range for heat exchanger j ($j = 1$, 2, or 3). If A_j is located in the first area range ($A_j \le 10\ m^2$), $y_{1j} = 1$, $y_{2j} = 0$, and $y_{3j} = 0$. Analogously, if A_j is located in the second area range ($10\ m^2 < A_j \le 25\ m^2$), $y_{1j} = 0$, $y_{2j} = 1$, and $y_{3j} = 0$. Finally, if A_j is located in the third area range ($25\ m^2 < A_j \le 50\ m^2$), $y_{1j} = 0$, $y_{2j} = 0$, and $y_{3j} = 1$.

The Logarithmic Mean Temperature Difference (LMTD) was calculated using Paterson's approximation [19]. The

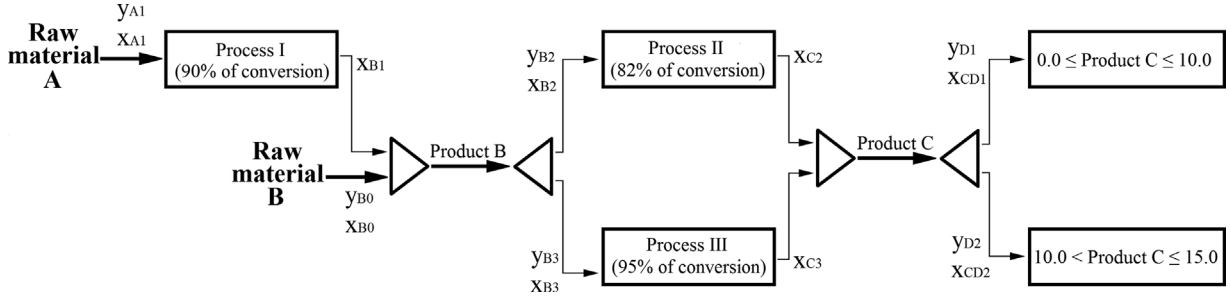

FIGURE 4 Superstructure for the production of chemical product C

continuous search variable T_1 is constrained between 351 and 500 K and the continuous search variable T_2 is constrained between 350 and 499 K, in accordance with the Second Law of Thermodynamics.

$$\begin{aligned}\min_{\mathbf{y},\mathbf{A},\mathbf{q},\mathbf{T}}\ \text{Cost} = {} & y_{11}(2750{A_1}^{0.6}+3000)+y_{21}(1500{A_1}^{0.6}+15000)+y_{31}(600{A_1}^{0.6}+46500)+\\ & {}_{y12}(2750{A_2}^{0.6}+3000)+y_{22}(1500{A_2}^{0.6}+15000)+y_{32}(600{A_2}^{0.6}+46500)+\\ & {}_{y13}(2750{A_3}^{0.6}+3000)+y_{23}(1500{A_3}^{0.6}+15000)+y_{33}(600{A_3}^{0.6}+46500)+\\ & 20q_{H\_LN\_2}+80q_{C\_LN\_3}\end{aligned}$$

$$\begin{aligned}s.t.\quad & y_{11}+y_{21}+y_{31}-1=0;\quad y_{12}+y_{22}+y_{32}-1=0;\quad y_{13}+y_{23}+y_{33}-1=0;\\ & T_{H\_inlet}-T_2-10\geq 0;\qquad T_1-T_{C\_inlet}-10\geq 0;\\ & A_1-10y_{11}-25y_{21}-50y_{31}\leq 0;\quad A_1-0y_{11}-10y_{21}-25y_{31}>0;\\ & A_2-10y_{12}-25y_{22}-50y_{32}\leq 0;\quad A_2-0y_{12}-10y_{22}-25y_{32}>0;\\ & A_3-10y_{13}-25y_{23}-50y_{33}\leq 0;\quad A_3-0y_{13}-10y_{23}-25y_{33}>0\end{aligned}\tag{7}$$

$$\begin{aligned}& \mathbf{y}=[y_{11},y_{12},y_{13},y_{21},y_{22},y_{23},y_{31},y_{32},y_{33}]^T\\ & \mathbf{A}=[A_1,A_2,A_3]^T,\ \mathbf{q}=[q_{H\_LN\_2},q_{C\_LN\_3}]^T\\ & \mathbf{T}=[T_1,T_2]^T\\ & \mathbf{y}\in\{0,1\}^9,\ \mathbf{A}\in\Re_+^3,\ \mathbf{q}\in\Re_+^2,\ \mathbf{T}\in\Re_+^2\\ & 0\leq A_j\leq 50,\ j\in\{1,2,3\}\\ & 351\leq T_1\leq 500\\ & 350\leq T_2\leq 499\end{aligned}$$

The solution of the described optimization problem was carried out using non-deterministic MINLP (MINLP-PSO) with 400 particles for 70 iterations, considering the default values for cognitive and social parameters and the inertia weight factor. The deterministic algorithm BONMIN with the B-BB (Branch-and-Bound) variation was also used, for results comparison. The heat exchanger model and the optimization Problem 7, written in the EMSO environment, are presented in the Appendix A. As in previous case study, the authors would like to highlight that, as far as they are concerned, no previous report of MINLP application in EMSO was found in the literature.

3.4 | Sugarcane biorefinery

A sugarcane biorefinery, producing sugar, bioethanol and bioelectricity, is used in order to demonstrate an application in EMSO of a multiobjective nonlinear optimization model in a highly integrated process. The modeled process was based on the standard first generation Brazilian process, which was described in the literature [6].

It considers a sugarcane input of 503 tonnes of sugarcane (TC) per hour. Sugarcane biorefinery block diagram is presented in Figure 8. In the process, sugarcane is washed prior to the sugarcane extraction operation. In the extraction step, the solid material is separated and sent to the boiler. Sugarcane juice is physically and chemically treated and sent to splitter 1, which is responsible for dividing sugarcane juice into two branches. One branch is sent to the concentration step in multiple-effect evaporators and the other is sent to fermentation step, to produce ethanol. Splitter 2, on the other hand, divide the stream of concentrated juice into two branches, one to be sent to fermentation step (ethanol production) and the other to be sent to crystallization step, aiming to produce sugar. In power generation step, steam and electrical power are produced, fulfilling process demands, and the surplus of electrical power can be sold, increasing the profitability of the process. Process steam consumption depends on the amount of sugar and ethanol produced. Details about this sugarcane biorefinery model are presented in Appendix B.

The relationships among ethanol and sugar production and electrical power generation were explored in three multiobjective optimization problems, described in the next sections. The proposed optimization problems have conflicting objective functions because it is not possible to simultaneously maximize ethanol and sugar production, since both are produced from the same source, sugarcane juice. Furthermore, changing production of ethanol or sugar affects the surplus of electrical power: maximizing any of the material products demands more energy (both thermal and electrical), which leads to a decrease in availability of electricity for sale. In all optimization problems the search

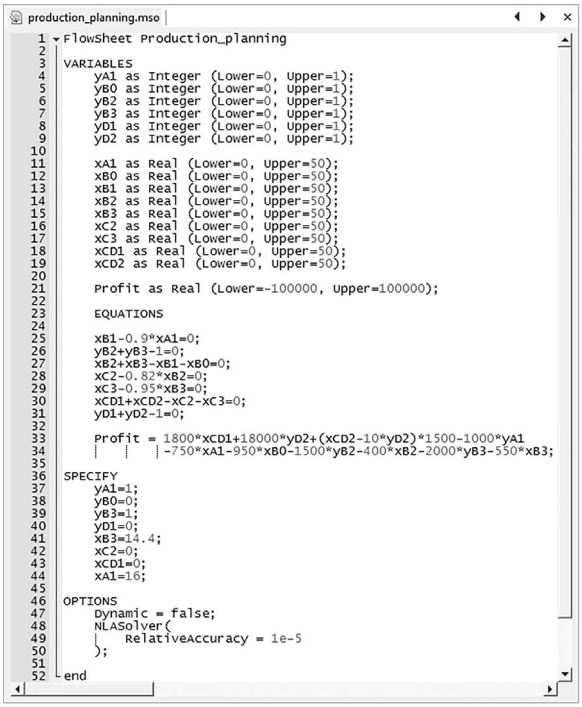

FIGURE 5 Flowsheet model for the production of chemical product C in EMSO software

variables are the split fractions in splitter 1 and 2. In this biorefinery, the model ensures that all steam and electrical power required by the process are provided by the power generation step and only the surplus of electrical power is sold. Furthermore, the proposed multiobjective optimization problems are nonlinear, as it is explicitly shown in equations in Appendix B. All problems were solved using MOO-PSO, with the default values for cognitive and social parameters and the inertia weight factor.

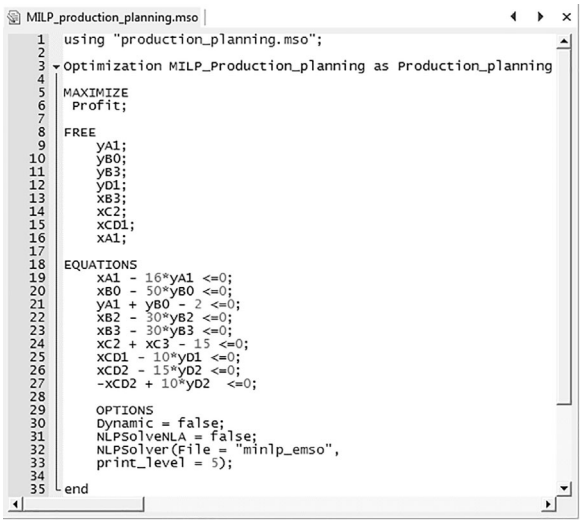

FIGURE 6 Production planning optimization problem in EMSO software

3.4.1 | Maximizing ethanol and sugar production

The first optimization study of the biorefinery was based on the objectives of maximizing both the production of ethanol by fermentation ($F_{ethanol}$) and sugar by crystallization (F_{conc_juice}) with the fractional variables (x_{SP1} and x_{SP2}) constrained between 0.2 and 0.8 for splitters 1 and 2. If the lower and upper bound to these variables are kept closer to zero and 1.0, respectively, indeterminacy in the models can occur during the optimization procedure. Problem 8 represents the optimization model. Note that Equations B.1 and B.3 bring nonlinearities to the optimization problem, since splitters fractions and mass flow rates are variables in the problem. MOO-PSO solver was used with 50 particles in 21 iterations.

$$\begin{aligned} &\max_{\mathbf{x_{SP}}} \left[F_{ethanol}(\mathbf{x_{SP}}), F_{conc\_juice}(\mathbf{x_{SP}})\right] \\ s.t. \quad &\text{Biorefinery model;} \\ &Eq.(B.1);\ Eq.(B.2);\ Eq.(B.3);\ Eq.(B.4); \\ &x_{SP1} - 0.8 < 0;\ 0.2 - x_{SP1} < 0;\ x_{SP2} - 0.8 < 0;\ 0.2 - x_{SP2} < 0 \\ &\mathbf{x_{SP}} = [x_{SP1}, x_{SP2}]^T \\ &\mathbf{x_{SP}} \in \Re_+^2 \end{aligned} \tag{8}$$

3.4.2 | Maximizing electrical power and ethanol production

Considering that the surplus of electrical power can be sold to the grid, this optimization problem aims to improve the electrical power production ($E_{electrical}$) and the ethanol production ($F_{ethanol}$) at the same time, as showed in Problem 9. As the steam produced by the boiler is supplied both to the process and to the turbines, a higher production of ethanol demands more thermal energy and, consequently, leads to less surplus of electrical power available for sale.

MOO-PSO solver was used with 50 particles in 11 iterations. For comparative purposes, this optimization was performed considering approximately half of the maximum number of iterations in relation to the previous optimization problem.

$$\begin{aligned} &\max_{\mathbf{x_{SP}}} \left[E_{electrical}(\mathbf{x_{SP}}), F_{ethanol}(\mathbf{x_{SP}})\right] \\ s.t. \quad &\text{Biorefinery model;} \\ &Eq.(B.1);\ Eq.(B.2);\ Eq.(B.3);\ Eq.(B.4);\ Eq.(B.5); \\ &Eq.(B.6);\ Eq.(B.7);\ Eq.(B.8);\ Eq.(B.9); \\ &x_{SP1} - 0.8 < 0;\ 0.2 - x_{SP1} < 0;\ x_{SP2} - 0.8 < 0;\ 0.2 - x_{SP2} < 0 \\ &\mathbf{x_{SP}} = [x_{SP1}, x_{SP2}]^T \\ &\mathbf{x_{SP}} \in \Re_+^2 \end{aligned} \tag{9}$$

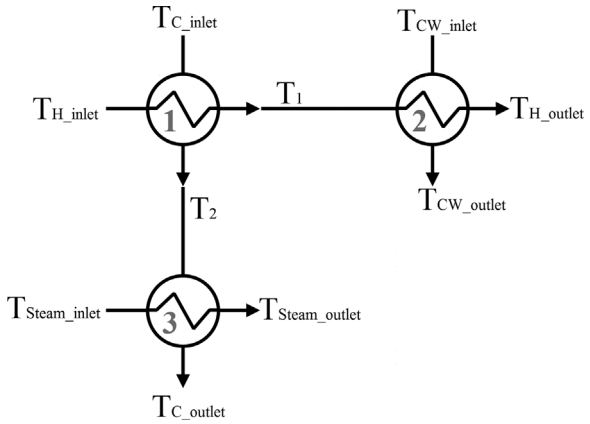

FIGURE 7 Installation scheme of heat exchanger network

3.4.3 | Maximizing electrical power and sugar production

According to Problem 10, the aim of this multobjective optimization is to maximize the surplus of electrical power and the production of concentrated juice. MOO-PSO solver was used with 50 particles in 21 iterations.

$$\max_{\mathbf{x_{SP}}} \left[E_{electrical}(\mathbf{x_{SP}}), F_{conc\_juice}(\mathbf{x_{SP}})\right]$$

$$\begin{aligned} &s.t. \\ &\quad \text{Biorefinery model;} \\ &\quad Eq.(B.1);\ Eq.(B.2);\ Eq.(B.3);\ Eq.(B.4);\ Eq.(B.5); \\ &\quad Eq.(B.6);\ Eq.(B.7)\ Eq.(B.8)\ Eq.(B.9); \\ &x_{SP1} - 0.8 < 0;\ 0.2 - x_{SP1} < 0;\ x_{SP2} - 0.8 < 0;\ 0.2 - x_{SP2} < 0 \\ &\mathbf{x_{SP}} = [x_{SP1}, x_{SP2}]^T \\ &\mathbf{x_{SP}} \in \Re^2_+ \end{aligned} \tag{10}$$

4 | RESULT AND DISCUSSIONS

To solve the case studies, EMSO software (version 0.10.6) and VRTherm software (1.5.0 version) were used in an AMD Phenom™ II X6 1090 Microcomputer Processor (3.20 GHz) with 8 GB RAM and graphics card GTX 760 with 2 GB in Linux Debian 7.6 environment (32-bit). VRTherm software is available in EMSO software installation package, containing more than 2,000 chemical compounds in the database and several thermodynamic models, which was used as a plugin to calculate the thermodynamic properties required in the biorefinery model.

4.1 | Continuous stirred tank reactors in series

The optimization problem of three CSTRs was solved with relative accuracy of 10^{-6} and the same value for the objective function was found in the solutions with the two different definitions of the Hessian matrix, with values 2.078, 2.332, and 2.590 m^3 for V_1, V_2, and V_3, respectively, and $C_{A3} = 0.210$ kmol/m^3. However, the trajectories during the search and the number of iterations were different, even with the same initial estimate. Considering the exact Hessian, the solver found the solution after eight iterations, in 0.016 s. Considering the limited-memory Hessian approximation, the solver found the solution after twenty iterations, in 0.052 s. It was possible to solve this optimization problem in less than 1 s because the problem consists of only 15 variables and possesses 3 degrees of freedom.

4.2 | Production planning

The optimization was solved in 0.256 s, and the solution, showed in Figure 9, demonstrates that the maximum profit is achieved if Processes I and III are used with 16 tonne/hr of A fed and no purchase of raw material B ($y_{B0} = 0$). With the purchase of 16 tonne/hr of A, 14.40 tonne/hr of B are produced in Process I. This amount is then entirely fed to Process III ($y_{B2} = 0$, $y_{B3} = 1$). The amount of component C produced is 13.68 tonne/hr, obtaining a profit of 600.00 $/hr. It was also verified that in the solution y_{D2} is equal to 1, and

TABLE 2 Heat exchanger network data (Adapted from Turkay and Grossmann [26])

Stream	F_{CP} (kW/K)	T_{inlet} (K)	T_{outlet} (K)	Cost ($/kW year)
Hot	10.0	500	340	-
Cold	7.5	350	560	-
Cooling water	-	300	320	20
Steam	-	600	600	80

Heat exchanger	Overall heat transfer coefficient (kW/[m^2K])	Heat exchanger	Feasible area (m^2)	Capital cost ($/year)
1	1.5	1, 2, and 3	$0 < A_j \leq 10$	$2{,}750\,A_j^{0.6} + 3{,}000$
2	0.5		$10 < A_j \leq 25$	$1{,}500\,A_j^{0.6} + 15{,}000$
3	1.0		$25 < A_j \leq 50$	$600\,A_j^{0.6} + 46{,}500$

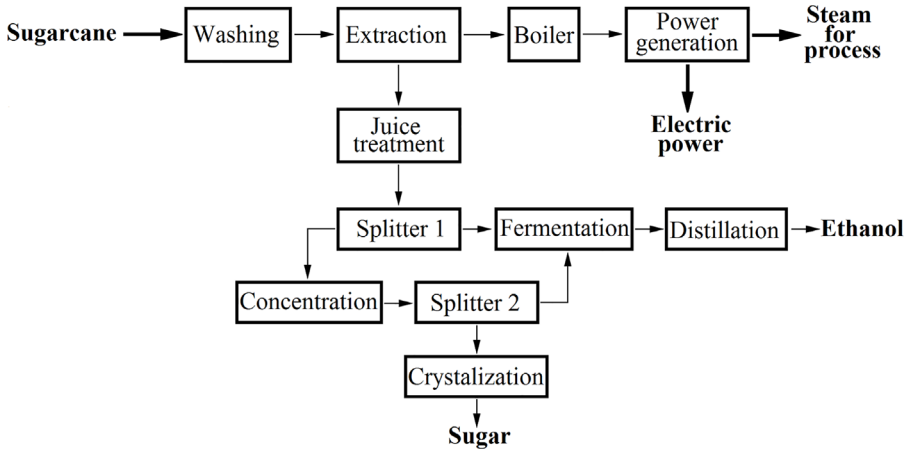

FIGURE 8 Process diagram of the sugarcane biorefinery

$y_{DI} = 0$, since the mass flow of C is smaller than the maximum demand (15 tonne/hr) and greater than 10 tonne/hr.

The computational time required to solve this problem with 8 degrees of freedom (15 variables and 7 equality constraints) was also very small. However, it was approximately 25 times greater than the time required to optimize the CSTR problem.

4.3 | Heat exchanger network

After performing the optimization runs, the results found by the two solvers (MINLP-PSO and BONMIN) were nearly identical (only small numerical differences). The optimal areas of heat exchangers 1, 2, and 3 are 25.00, 20.18, and 7.68 m^2, respectively. With these areas, the operating condition determines $T_1 = 397.3$ K and $T_2 = 486.9$ K, generating a total annual cost of 117,105.40 \$/year. Although Turkay and Grossmann [26] did not use the Paterson's approximation to calculate the LMTD, which could underestimate heat exchange areas, the results reported are very similar and the areas of the same heat exchangers are 25.00, 20.35, and 7.85 m^2, generating a total annual cost of 117,781.00 \$/year.

The deterministic code demanded 0.528 s, while the non-deterministic one required 5.376 s for solving this problem, which presents 33 variables and 7 degrees of freedom. Naturally, this difference is related to the use of 400 particles and 70 iterations in the non-deterministic solver. Since both methods were able to find the same optimum, there is no gain in using the non-deterministic solver. However, for larger problems with local minima, this solver may be advantageous. Comparing the production planning optimization and this optimization problem, it can be verified that there is an increase in the number of variables from 15 to 33 and this relation implies in the computational time increase from 0.256 s (production planning problem) to 0.528 s in this optimization problem.

4.4 | Sugarcane biorefinery

4.4.1 | Maximizing ethanol and sugar production

The computational time for solving this problem, with 1,307 variables, 1,256 equations, 4 inequality constraints and 2 degrees of freedom, was 68.88 s. Figure 10a shows the non-dominated solutions in the functions space. A linear profile between the two objective functions is observed. This relationship can be explained in terms of process because for a higher production of ethanol the flow rate of

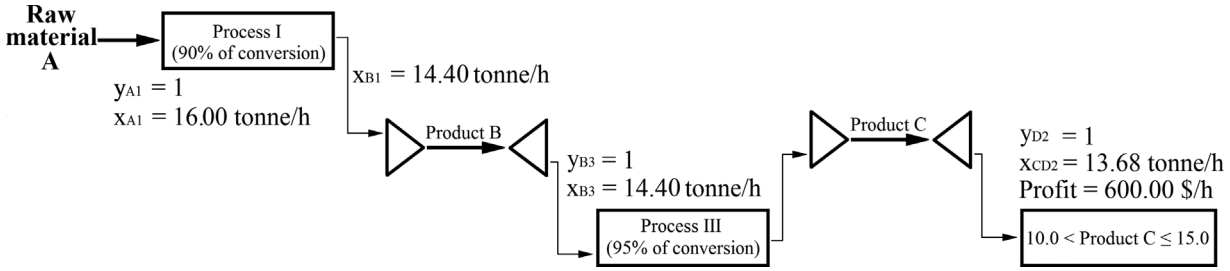

FIGURE 9 Optimal structure for the production planning, with values for the decision variables and objective function

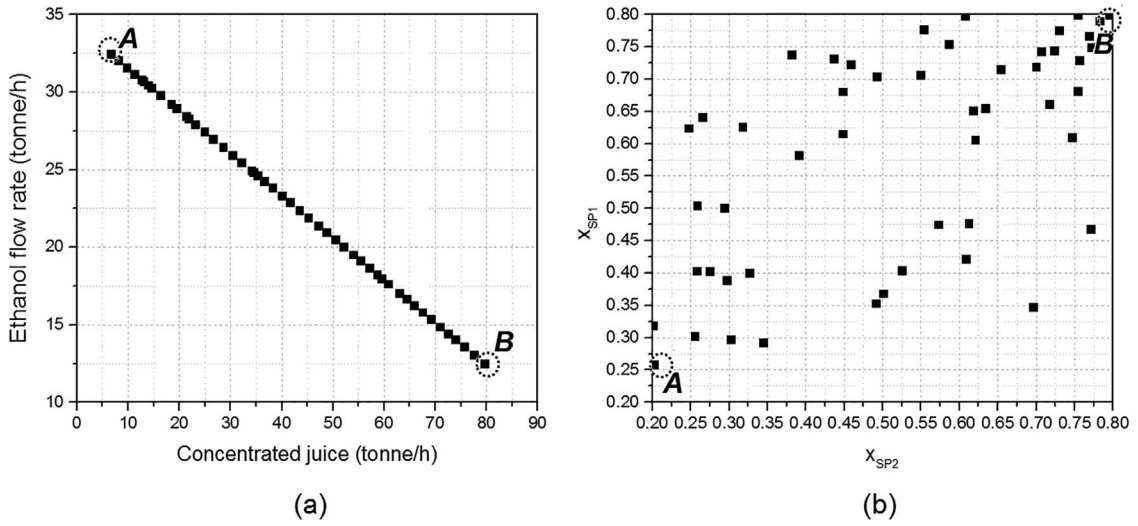

FIGURE 10 Non-dominated solutions and decision variables for multiobjective optimization of ethanol flow rate and concentrated juice

concentrated juice sent to the crystallization step must be reduced. In Figure 10b, it is possible to observe that non-dominated solutions are spread in the search space. The continuous decision variables x_{SP1} and x_{SP2} present a distribution in the range of 0.258–0.799 and 0.201–0.795, respectively. In the extreme down-right region, the particles are viable but are dominated by other particles that are presented in the graph. For example, with an operating condition of $x_{SP1} = 0.201$ and $x_{SP2} = 0.799$, ethanol production is 28.731 tonne/hr and concentrated juice mass flow rate is 20.130 tonne/hr, which is a dominated solution. Figure 10a shows that is possible to produce this same throughput of sugar (20.130 tonne/hr) and simultaneously produce more ethanol (28.752 tonne/hr).

Point A, presented in Figure 10, represents the solution with maximum ethanol production (32.453 tonne/hr) and minimum concentrated juice production (6.551 tonne/hr). In this case, the continuous variables x_{SP1} and x_{SP2} have values of 0.258 and 0.202, respectively. On the other hand, point B represents the operating condition in which the concentrated juice production is maximized (79.660 tonne/hr) and ethanol production is minimum (12.487 tonne/hr). This condition is reached when the continuous variables x_{SP1} and x_{SP2} have values of 0.799 and 0.795, respectively. Changing from the operating condition of point A to that of point B implies a great increase in concentrated juice production (1115.96%) and a not overmuch decrease in ethanol production (61.52%).

4.4.2 | Maximizing electrical power and ethanol production

The computational time required to solve this problem, that possesses the same number of variables (1,307) and degrees of freedom (2) as the previous one, was 33.43 s, nearly half of the required in that problem, due to the reduced number of iterations.

Figure 11a shows the Pareto front of the non-dominated solutions, and a nonlinear profile between the two objective functions is observed. This relationship can be explained in terms of the nonlinearity of boiler and turbine models of the cogeneration system and the nonlinearity of multiple-effect evaporators and fermentation models. According to Figure 11b, x_{SP2} presents a distribution between 0.200 and 0.784, while x_{SP1} ranges narrowly, between 0.243 and 0.253. This narrow distribution results in a small variation, among non-dominated particles, of the energy required by the multiple-effect evaporators (between 18.96 and 19.50 MW). Most of treated sugarcane juice is sent to fermentation stage, since sugarcane production is not set as a goal. Even though the values of x_{SP2} have great variation, in all non-dominated particles the large amount of treated juice sent to fermentation results in a variation of the surplus of electricity from 49.86 to 50.11 MW, between extreme non-dominated solutions (points C and D, respectively). For the sake of comparison, in previous optimization study, for which electrical power surplus was not set as one of the objective functions, electrical power available for sale varied between 49.82 MW (point A, maximum of ethanol production) and 47.63 MW (point B, maximum concentrated juice mass flow rate). The fact that x_{sp1} is narrowly distributed among non-dominated particles leads to a small variation of ethanol flow between points C and D (approximately 4.85 tonne/hr).

In this multi-objective optimization, point D presents the operating condition of maximum electrical power generation and minimum ethanol production, when x_{SP1} and x_{SP2} have values of 0.245 and 0.784, respectively. However, in point C,

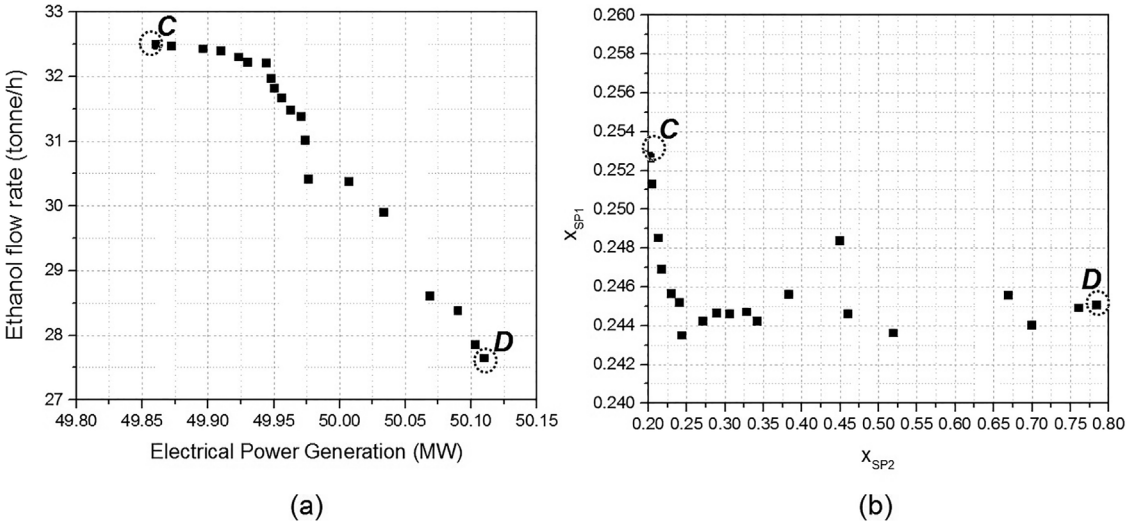

FIGURE 11 Non-dominated solutions and decision variables for multiobjective optimization of ethanol flow rate and electrical power generation

when the ethanol production is maximized, the electrical power generation is minimum. This point is obtained when x_{SP1} and x_{SP2} have values of 0.253 and 0.201, respectively. Between points C and D, electrical power varies 0.50%, while ethanol production varies 14.93%.

4.4.3 | Maximizing electrical power and sugar production

The computational time required to solve this problem was approximately 65.24 s, very close to that necessary in the first optimization study in the biorefinery (maximization of ethanol and sugar production), since the computational time is most related to the size of the problem, number of particles and number of iterations.

Figure 12a shows the Pareto front of the non-dominated solutions, which is clearly non-linear, due to the same reasons previously observed. According to Figure 12b, x_{SP1} presents a distribution between 0.246 and 0.799, and x_{SP2} varies between 0.774 and 0.796. This large variation, among non-dominated solutions, of the search variable x_{SP1} and small variation of the search variable x_{SP2} shows that the evaporation process has a great influence on the electrical energy available for sale. In this optimization problem, with the objectives of maximizing electricity generation and sugar production, solutions with large values for x_{SP1} and small

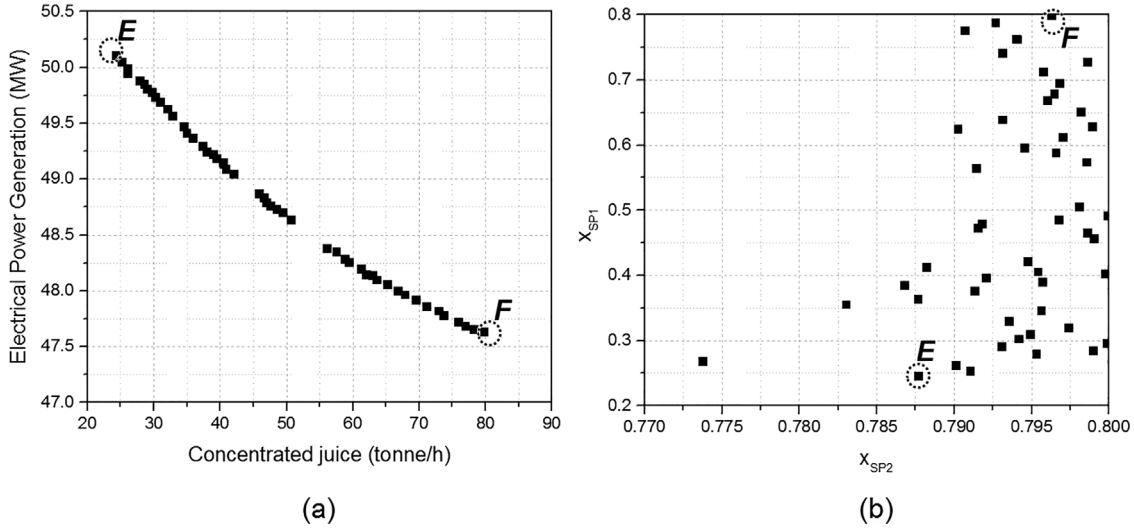

FIGURE 12 Non-dominated solutions and decision variables for multiobjective optimization of electrical power generation and concentrated juice

values for x_{SP2} are dominated by the particles shown in Figure 12b. For example, with an operating condition of $x_{SP1} = 0.799$ and $x_{SP2} = 0.201$, electricity surplus is 46.99 MW and concentrated juice production is 20.13 tonne/hr and, according to Figure 12a, this is a dominated operating condition.

It is possible to verify a large variation of the concentrated juice production, between 24.25 and 79.76 tonne/hr, and a variation of approximately 2.5 MW in the surplus of electrical power, between 47.63 e 50.11 MW. In other words, between the two extreme solutions (points E and F), while the concentrated juice production F_{conc_juice} increases 228.9%, the electricity generation $E_{electrical}$ decreases 4.94%.

The operating condition of maximum electrical power generation and minimum concentrated juice production showed in point E of Figure 12 is obtained when x_{SP1} and x_{SP2} have values of 0.246 and 0.788, respectively. However, in point F, when the concentrated juice production is maximized, the electrical power generation is minimum. This point is obtained when x_{SP1} and x_{SP2} have values of 0.799 and 0.796, respectively.

5 | CONCLUSIONS

The solution of different types of chemical engineering optimization problems demonstrated the great versatility of EMSO software. It was possible to solve simple and complex optimization problems in this software obtaining the same literature results with very short computational time. In this paper, optimization problems ranging from 15 to 1,307 variables were solved, demanding up to 68.88 s, considering deterministic and nondeterministic methods. Faced with this great versatility, ease of model development and applicability, EMSO software shows itself capable of being used in optimization courses in undergraduate and postgraduate programs in chemical or process systems engineering.

ACKNOWLEDGMENTS

The authors thank the financial support provided by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) (Process 132162/2015-6), and FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) (Process 2011/51902-9). Also, the authors thank Felipe Fernando Furlan for kindly providing support with EMSO models and with MOO-PSO plugin.

NOMENCLATURE

Parameters

C_{A0}	component A concentration in the feed of Reactor 1 (kmol/m^3)
C_{Ai}	component A concentration in the outlet of Reactor i, $i \in \{1,2,3\}$ (kmol/m^3)
F_0	constant volumetric flow (m^3/hr)
F_{CP_C}	cold stream total heat capacity (kW/K)
F_{CP_H}	hot stream total heat capacity (kW/K)
k	reaction constant (m$^{1.5}$/[hr kmol$^{0.5}$])
T_{C_inlet}	inlet temperature of cold stream in heat exchanger 1 (K)
T_{C_outlet}	outlet temperature of cold stream from heat exchanger 3 (K)
T_{CW_inlet}	inlet temperature of cooling water in heat exchanger 2 (K)
T_{CW_outlet}	outlet temperature of cooling water from heat exchanger 2 (K)
T_{H_inlet}	inlet temperature of hot stream in heat exchanger 1 (K)
T_{H_outlet}	outlet temperature of hot stream from heat exchanger 2 (K)
T_{Steam_inlet}	inlet temperature of steam in heat exchanger 3 (K)
T_{Steam_outlet}	outlet temperature of steam from heat exchanger 3 (K)
U_i	overall heat transfer coefficient of heat exchanger i, $i \in \{1,2,3\}$ (kW/m^{2}K)

Continuous variables

A_j	heat transfer area of heat exchanger j, $j \in \{1,2,3\}$ (m^2)
c_1, c_2	cognitive and social parameters, respectively
$E_{electrical}$	surplus of electrical power (MW)
$E_{electrical\_total}$	total power generated by the four turbines (MW)
$E_{distillery}$	electrical energy consumed by the distillery (MW)
$E_{biorefinery}$	electrical energy consumed by the biorefinery (MW)
F_{1_mixer}	inlet stream 1 in mixer (tonne/hr)
F_{2_mixer}	inlet stream 2 in mixer (tonne/hr)
F_{conc_juice}	mass flow of concentrated juice produced (65 °Brix) (tonne/hr)
$F_{effect1}$	inlet stream in evaporator 1 (tonne/hr)
$F_{ethanol}$	mass flow of ethanol produced in the fermentation stage (tonne/hr)

F_{in_T1} inlet of superheated steam in turbine 1 (tonne/hr)
F_{out_T1} mass flow of outlet superheated steam from turbine 1 (tonne/hr)
F_{out_T2} mass flow of outlet superheated steam from turbine 2 (tonne/hr)
F_{out_T3} mass flow of outlet superheated steam from turbine 3 (tonne/hr)
F_{out_TM} mass flow of outlet superheated steam from turbine M (tonne/hr)
F_{outs_T1} mass flow of outlet heated steam from turbine 1 (tonne/hr)
F_{outs_T2} mass flow of outlet heated steam from turbine 2 (tonne/hr)
F_{SP1} treated juice mass flow sent to evaporator (tonne/hr)
F_{SP2} concentrated juice mass flow sent to sugar production (tonne/hr)
n number of particles in the swarm
P profit of a supplier of chemical component C ($/hr)
Pot_T1 power generated in turbine 1 (MW)
Pot_T2 power generated in turbine 2 (MW)
Pot_T3 power generated in turbine 3 (MW)
Pot_TM power generated in turbine M (MW)
$q_{H\_LN\_j}$ heat load from hot stream in heat exchanger j, $j \in \{1,2,3\}$ (kW)
$q_{C\_LN\_j}$ heat load from cold stream in heat exchanger j, $j \in \{1,2,3\}$ (kW)
Q_{steam} steam required by the biorefinery process (MW)
$Q_{juice\_preparation}$ steam required by the five effect evaporator (MW)
$Q_{evaporators}$ steam required for juice preparation (MW)
r_{Ai} reaction rate of component A in reactor i, $i \in \{1,2,3\}$ (kmol/[hr m^3])
T_1 outlet temperature of hot stream from heat exchanger 1 (K)
T_2 outlet temperature of cold stream from heat exchanger 1 (K)
V_i volume of reactor i, $i \in \{1,2,3\}$ (m^3)
w inertia weight factor
x_{Ai} mass flow of raw material A (tonne/hr)
x_{B0} mass flow of raw material B (tonne/hr)
x_{B1} mass flow of B produced by Process I (tonne/hr)
x_{B2} mass flow of Product B diverted to Process II (tonne/hr)
x_{B3} mass flow of Product B diverted to Process III (tonne/hr)
x_{C2} mass flow of C produced by Process II (tonne/hr)
x_{C3} mass flow of C produced by Process III (tonne/hr)
x_{CD1} mass flow of Product C up to 10.0 tonne/hr (tonne/hr)
x_{CD2} mass flow of Product C greater than 10.0 and less than15.0 tonne/hr (tonne/hr)
x_{SP1} fraction of treated juice stream sent to evaporator (splitter 1)
x_{SP2} fraction of concentrated juice stream sent to sugar production (splitter 2)
x_{T1} fraction of inlet superheated steam in turbine 1 sent to turbine M
x_{T2} fraction of inlet steam in turbine 2 sent to splitter 3

Binary variables

y_{A1} equal to 1 if raw material A is purchased, and 0 otherwise
y_{B0} equal to 1 if raw material B is purchased, and 0 otherwise
y_{B2} equal to 1 if Process II is chosen, and 0 otherwise
y_{B3} equal to 1 if Process III is chosen, and 0 otherwise
y_{D1} equal to 1 if Product C flow is between 0 and 10.00, and 0 otherwise
y_{D2} equal to 1 if the Product C flow is greater than 10.00 and lower than or equal to 15.00, and 0 otherwise
y_{ij} binary variable related to area range i of heat exchanger j

ORCID

João Paulo Henrique http://orcid.org/0000-0001-5550-1441
Ruy de Sousa Jr http://orcid.org/0000-0003-4916-173X
Argimiro R. Secchi http://orcid.org/0000-0001-7297-3571
Mauro A. S. S. Ravagnani http://orcid.org/0000-0002-2151-1534
Caliane B. B. Costa http://orcid.org/0000-0002-9983-566X

REFERENCES

1. L. T. Biegler, I. E. Grossmann, and A. W. Westerberg, *Systematic methods of chemical process design* (1st ed.), Prentice Hall, New Jersey, 1997, p. 796.
2. P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and

A. Wätcher, *An algorithmic framework for convex mixed integer nonlinear programs*, Discrete. Optim. **5** (2008), 186–204.
3. J. A. Caballero, *Logic hybrid simulation-optimization algorithm for distillation design*, Comp. Chem. Eng. **72** (2015), 284–299.
4. C. B. B. Costa, E. Potrich, and A. J. G. Cruz, *Multiobjective optimization of a sugarcane biorefinery involving process and environmental aspects*, Renew. Energ. **96** (2016), 1142–1152.
5. F. C. Cunha, M. B. De Souza, A. G. Barreto, and D. F. S. De Souza, *Dynamic simulation of a compressor located in a natural gas processing unit using EMSO simulator*, Comp. Aid. Chem. Eng. **27** (2009), 603–608.
6. M. O. S. Dias, A. V. Ensinas, S. A. Nebra, R. Maciel Filho, C. E. V. Rossell, and M. R. W. Maciel, *Production of bioethanol and other bio-based materials from sugarcane bagasse: Integration to conventional bioethanol production process*, Chem. Eng. Res. Des. **87** (2009), 1206–1216.
7. A. Dimian, *Integrated design and simulation of chemical processes*, Comp. Aid. Chem. Eng. **13** (2003), 1–698.
8. M. A. Duran and I. E. Grossmann, *An outer approximation algorithm for a class of mixed-integer nonlinear programs*, Math. Prog. **36** (1986), 307–339.
9. M. Errico, B. G. Rong, C. E. Torres-Ortega, and J. G. Segovia-Hernandez, *The importance of the sequential synthesis methodology in the optimal distillation sequences design*, Comp. Chem. Eng. **62** (2014), 1–9.
10. C. A. Floudas, *Nonlinear and mixed-integer optimization* (1st ed.), Oxford University Press, New Jersey, 1995, p. 480.
11. F. F. Furlan, C. B. B. Costa, G. C. Fonseca, R. P. Soares, A. R. Secchi, A. J. G. Cruz, and R. C. Giordano, *Assessing the production of first and second generation bioethanol from sugarcane through the integration of global optimization and process detailed modeling*, Comp. Chem. Eng. **43** (2012), 1–9.
12. F. F. Furlan, A. R. A. Lino, K. Matugi, A. J. G. Cruz, A. R. Secchi, and R. C. Giordano, *A simple approach to improve the robustness of equation-oriented simulators: Multilinear look-up table interpolators*, Comp. Chem. Eng. **86** (2016), 1–4.
13. L. C. G. Gonçales, F. F. Furlan, R. P. Soares, A. R. Secchi, R. C. Giordano, and C. B. B. Costa, *Implementation of pareto multi-objective particle swarm optimization algorithm in EMSO*, EngOpt **73** (2012), 1–7.
14. O. K. Gupta, *Branch and bound experiments in nonlinear integer programming*. PhD Thesis, School of Industrial Engineering, Purdue University, 1980.
15. J. Kennedy and R. C. Eberhart, Particle swarm optimization. In IEEE International conference on neural networks (ICNN), 1995, 1942–1948.
16. J. Kennedy and R. C. Eberhart, *A discrete binary version of the particle swarm algorithm*, IEEE Int. Conf. Syst. Man. Cybern. **05** (1997), 4104–4108.
17. E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, *Particle swarm optimization for integer programming*, IEEE C. Evol. Comp. **02** (2002), 1582–1587.
18. J. Ospino, M. E. Sánchez, and A. R. Secchi, *Implementation of a block-oriented model library for undergraduate process control courses in EMSO simulator*, Educ. Chem. Eng. **1** (2016), 1–13.
19. W. R. Paterson, *A replacement for the logarithmic mean*, Chem. Eng. Sci. **39** (1984), 1635–1636.
20. I. Quesada and I. E. Grossmann, *An LP/NLP based branched and bound algorithm for convex MINLP optimization problems*, Comp. Chem. Eng. **16** (1992), 937–947.
21. C. R. Raquel and P. C. Naval Jr., *An effective use of crowding distance in multiobjective particle swarm optimization*, GECCO 05 **1** (2005), 257–264.
22. J. B. Restrepo, G. Olivar, and C. A. Cardona, *Bifurcation analysis of dynamic process models using Aspen Dynamics® and Aspen Custom Modeler®*, Comp. Chem. Eng. **62** (2014), 10–20.
23. R. Rodrigues, R. P. Soares, and A. R. Secchi, *Teaching chemical reaction engineering using EMSO simulator*, Comp. Appl. Eng. Educ. **18** (2009), 607–618.
24. R. P. Soares, *Emso Reference Manual*, 2007. Available at: http://www.enq.ufrgs.br/trac/alsoc, accessed on 5th January, 2017.
25. R. P. Soares and A. R. Secchi, *EMSO: A new environment for modelling, simulation and optimization*, Comp. Aid. Chem. Eng. **14** (2003), 947–952.
26. M. Turkay and I. E. Grossmann, *Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs—Multiple size regions*, J. Ind. Eng. Chem. **35** (1996), 2611–2623.
27. A. Valleriote, L. Dorigo, A. R. Secchi, and E. C. Biscaia Jr., *Reduced rigorous models for efficient dynamic simulation and optimization of distillation columns*, Comp. Aid. Chem. Eng. **30** (2012), 1262–1266.
28. A. Wächter, Short tutorial: Getting started with Ipopt in 90 minutes. *Combinatorial scientific computing*. Dagstuhl Seminar Proceedings 09061, Dagstuhl, Germany, 2009.
29. A. Wächter, *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD Thesis, Carnegie Mellon University, Pittsburgh, USA, 2002, p. 229.
30. A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program. **106** (2006), 25–57.

J. P. HENRIQUE bachelor's at Food Engineering from Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas (Ibilce), Câmpus São José do Rio Preto (2013), master's at Chemical Engineering from Universidade Federal de São Carlos (UFSCar), Câmpus São Carlos (2017). Currently is doctoral student at Food Engineering from Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas (Ibilce), Câmpus São José do Rio Preto. His major fields of interest are design, development and simulation of solid-state fermentation bioreactors, solid-state fermentation for enzyme production, biofuel production, process control and optimization, process modeling and simulation.

R. de Sousa Jr. is a professor of Chemical Engineering at the Department of Chemical Engineering, Federal University of São Carlos (UFSCar), Brazil, since 2010. Prior to that, he was postdoctoral researcher at the São Carlos Institute of Chemistry–University of São Paulo, Brazil (from 2005 to 2009), and at the Laboratoire de Catalyse en Chimie Organique-Université de Poitiers, France (in 2006). He received a PhD in Chemical Engineering, Graduate Program in Chemical Engineering-UFSCar, Brazil, 2003; MSc in Chemical Engineering, Graduate Program in Chemical Engineering-UFSCar, Brazil, 1999; and a BSc in Chemical Engineering, School of Chemical Engineering-University of Campinas, Brazil, 1996. His research interests mainly focus on modeling, simulation, control and optimization of biochemical, and electrochemical processes.

A. R. Secchi bachelor's at Engenharia Química from Universidade Federal do Rio Grande do Sul (1986), master's at Chemical Engineering from Universidade Federal do Rio de Janeiro (1988) and doctorate at Chemical Engineering from Universidade Federal do Rio de Janeiro in collaboration with CalTech (1992). Currently is full professor at COPPE - Universidade Federal do Rio de Janeiro. His major fields of interest are process modeling and simulation, process control and optimization, numerical methods for dynamic simulation and optimization, and real-time monitoring and optimization.

M. A. S. S. Ravagnani PhD in Chemical Engineering, Universidade Estadual de Campinas, Unicamp, Brazil (1994). During 2005–2007 he developed post-doctoral studies at the Universidad de Alicante (UA), Spain. He was director of the Technology Center at UEM (1996–2000 and 2008–2010), vice-rector for Research and Graduate Studies at the Universidade Estadual de Maringá (UEM) , Brazil (2010–2014) and president of the FOPROP (Forum of vice-rectors of research and graduate studies of the Brazilian universities). Currently he is professor of Graduate Studies Program in Chemical Engineering and of Chemical Engineering Department at UEM. He has experience in Chemical Engineering, focusing on Unit Operations and Chemical Engineering equipment, acting on the following subjects optimisation, minimization of energy and water consumption in industrial processes, pinch analysis, heat exchanger networks, process synthesis, process design, shell and tube heat exchangers and heat integration.

C. B. B. Costa received her BS, MS and PhD degrees in Chemical Engineering from Universidade Estadual de Campinas (UNICAMP), Brazil, in 1999, 2003, and 2006, respectively. From 2009 until 2015, she was a professor at Chemical Engineering Department at Universidade Federal de São Carlos (UFSCar), Brazil. Since 2015, she is a professor of the Chemical Engineering Department at Universidade Estadual de Maringá (UEM), Brazil. Her research interests are in the process systems engineering area, with applications in process synthesis, process modeling and simulation, process control and optimization.

How to cite this article: Henrique JP, de Sousa Jr. R, Secchi AR, Ravagnani MASS, Costa CBB. Optimization of chemical engineering problems with EMSO software. *Comput Appl Eng Educ*. 2017;1–21.
https://doi.org/10.1002/cae.21867

APPENDIX A

Heat Exchanger Network Models

To develop this model in the EMSO environment, user should create a file named *heat_exchanger.mso* and add the following commands.

```
using "types";
binary as Integer (Lower=0, Upper=1);

FlowSheet Heat_exchanger_network

  PARAMETERS

    Fcp_H   as Real (Default=10);  #Hot stream total heat capacity [KW/K]
    Fcp_C   as Real (Default=7.5); #Cold stream total heat capacity [KW/K]
    TH_in   as Real (Default=500); #Inlet temperature of hot stream in heat exchanger 1 [K]
    TC_in   as Real (Default=350); #Inlet temperature of cold stream in heat exchanger 1 [K]
    TH_out  as Real (Default=340); #Outlet temperature of hot stream from heat exchanger 2 [K]
    TCW_in  as Real (Default=300); #Inlet temperature of cooling water in heat exchanger 2 [K]
    TCW_out as Real (Default=320); #Outlet temperature of cooling water from heat exchanger 2 [K]
    TS_in   as Real (Default=600); #Inlet temperature of steam in heat exchanger 3 [K]
    TS_out  as Real (Default=600); #Outlet temperature of steam from heat exchanger 3 [K]
    TC_out  as Real (Default=560); #Outlet temperature of cold stream from heat exchanger 3 [K]
    U1      as Real (Default=1.5); #Overall heat transfer coefficient of heat exchanger 1 [KW/m2.K]
    U2      as Real (Default=0.5); #Overall heat transfer coefficient of heat exchanger 2 [KW/m2.K]
    U3      as Real (Default=1.0); #Overall heat transfer coefficient of heat exchanger 3 [KW/m2.K]

    VARIABLES

    T1      as positive (Default=370, Lower=351, Upper=500); #Outlet temperature of hot stream
                                    #      from heat exchanger 1 [K]
    T2      as positive (Default=450, Lower=350, Upper=499); #Outlet temperature of cold stream
                            #      from heat exchanger 1 [K]
    A1      as positive (Lower=0, Upper=50); #Heat transfer area of heat exchanger 1 [m2]
    A2      as positive (Lower=0, Upper=50); #Heat transfer area of heat exchanger 2 [m2]
    A3      as positive (Lower=0, Upper=50); #Heat transfer area of heat exchanger 3 [m2]

    e_troc1       as Real;
    tetaAM_troc1  as Real;
    tetaGM_troc1  as Real;
    deltaT_ln_1   as Real; #LMTD at Heat exchenger 1 by PATERSON's Method, 1984
    e_troc2       as Real;
    tetaAM_troc2  as Real;
    tetaGM_troc2  as Real;
    deltaT_ln_2   as Real; #LMTD at Heat exchenger 1 by PATERSON's Method, 1984
    e_troc3       as Real;
    tetaAM_troc3  as Real;
    tetaGM_troc3  as Real;
    deltaT_ln_3   as Real; #LMTD at Heat exchenger 1 by PATERSON's Method, 1984

    qH_LN_1       as Real; #Heat load from hot stream in heat exchanger 1, obtained by PATERSON's Method
    qH_CP_1       as Real; #Heat load from hot stream in heat exchanger 1
    qH_LN_2       as Real; #Heat load from hot stream in heat exchanger 2, obtained by PATERSON's Method
    qH_CP_2       as Real; #Heat load from hot stream in heat exchanger 2
    qC_LN_3       as Real; #Heat load from cold stream in heat exchanger 3, obtained by PATERSON's Method
    qC_CP_3       as Real; #Heat load from cold stream in heat exchanger 3

    y11 as binary;
    y12 as binary;
    y13 as binary;
```

```
        y21 as binary;
        y22 as binary;
        y23 as binary;
        y31 as binary;
        y32 as binary;
        y33 as binary;

        Cost as Real;

        EQUATIONS

        e_troc1 = 1 - (TH_in-T2)/(T1-TC_in);
        e_troc2 = 1 - (TH_out-TCW_in)/(T1-TCW_out);
        e_troc3 = 1 - (TS_in-TC_out)/(TS_out-T2);
        tetaAM_troc1 = (T1-TC_in)*(1-(e_troc1)/2);
        tetaGM_troc1 = (T1-TC_in)*(1-(e_troc1)/2-(e_troc1^2)/8-(e_troc1^3)/16);
        tetaAM_troc2 = (T1-TCW_out)*(1-(e_troc2)/2);
        tetaGM_troc2 = (T1-TCW_out)*(1-(e_troc2)/2-(e_troc2^2)/8-(e_troc2^3)/16);
        tetaAM_troc3 = (TS_out-T2)*(1-(e_troc3)/2);
        tetaGM_troc3 = (TS_out-T2)*(1-(e_troc3)/2-(e_troc3^2)/8-(e_troc3^3)/16);
        deltaT_ln_1 = (1/3)*tetaAM_troc1 + (2/3)*tetaGM_troc1;
        deltaT_ln_2 = (1/3)*tetaAM_troc2 + (2/3)*tetaGM_troc2;
        deltaT_ln_3 = (1/3)*tetaAM_troc3 + (2/3)*tetaGM_troc3;
        qH_LN_1 - U1*A1*deltaT_ln_1 = 0;
        qH_CP_1 - Fcp_H*(TH_in-T1) = 0;
        qH_LN_1 - qH_CP_1 = 0;
        qH_LN_2 - U2*A2*deltaT_ln_2 = 0;
        qH_CP_2 - Fcp_H*(T1-TH_out) = 0;
        qH_LN_2 - qH_CP_2 = 0;
        qC_LN_3 - U3*A3*deltaT_ln_3 = 0;
        qC_CP_3 - Fcp_C*(TC_out-T2) = 0;
        qC_LN_3 - qC_CP_3 = 0;

        qH_LN_1 - Fcp_C*(T2-TC_in)=0;

        y11+y21+y31-1=0;
        y12+y22+y32-1=0;
        y13+y23+y33-1=0;

        Cost = y11*(2750*(A1^0.6)+3000) + y21*(1500*(A1^0.6)+15000) + y31*(600*(A1^0.6)+46500) +
            y12*(2750*(A2^0.6)+3000) + y22*(1500*(A2^0.6)+15000) + y32*(600*(A2^0.6)+46500) +
            y13*(2750*(A3^0.6)+3000) + y23*(1500*(A3^0.6)+15000) + y33*(600*(A3^0.6)+46500) +
            20*qH_LN_2 + 80*qC_LN_3;

        SPECIFY
        T1=398.15;
        y11=0;
        y21=1;
        y12=0;
        y22=1;
        y13=1;
        y23=0;

    OPTIONS
        Dynamic = false;
        NLASolver(
        RelativeAccuracy = 1e-5
        );

end
```

To develop the heat exchanger network optimization problem (Problem 7) in the EMSO environment, user should create a file named *MINLP_heat_exchanger.mso* in the same folder of the previous file and add the following commands.

```
using "heat_exchanger.mso";

Optimization MINLP_Heat_exchanger_network as Heat_exchanger_network

        EQUATIONS
        TH_in - T2 >=10;
        T1 - TC_in >=10;
        A1 - 10*y11 - 25*y21 - 50*y31 <= 0;
        A1 - 0*y11 - 10*y21 - 25*y31 > 0;
        A2 - 10*y12 - 25*y22 - 50*y32 <= 0;
        A2 - 0*y12 - 10*y22 - 25*y32 > 0;
        A3 - 10*y13 - 25*y23 - 50*y33 <= 0;
        A3 - 0*y13 - 10*y23 - 25*y33 > 0;

        MINIMIZE
        Cost;

        FREE
        T1;
        y11;
        y12;
        y13;
        y21;
        y22;
        y23;

        OPTIONS
        Dynamic = false;
        NLPSolveNLA = false;

        #-------------------------------------------------------------#
        #--------------------MINLP_Deterministic---------------------#
        #-------------------------------------------------------------#
        NLPSolver(File = "minlp_emso",
        bonmin_algorithm = "B-BB",
        print_level = 5);

        #-------------------------------------------------------------#
        #-------------------------MINLP_PSO---------------------------#
        #-------------------------------------------------------------#
        #*
        NLPSolver(File = "MINLP-PSO",
           verbosity = 4,
                   MaxIterations = 70,
                   number_of_particles = 400
                   );

        GuessFile="Heat_exchanger_network.rlt";

        FeasiblePath = true;
   *#

end
```

APPENDIX B

Sugarcane Biorefinery Model Details

In order to complement the biorefinery information presented in Case studies Section, Figure B1 illustrates the concentration step and the division of treated juice into two branches (one for the evaporators and the other to the fermentation step). Splitter 1, represented by Equations B.1 and B.2, may divert part of the juice to evaporation in order to concentrate it to 65°Brix (i.e., to a mass fraction of sugar of 0.65). This concentrated juice can be entirely processed into sugar (F_{conc_juice}) or can be partially diverted to ethanol production (F_{2_mixer}), through splitter 2, represented by Equations B.3 and B.4.

$$x_{SP1}(F_{SP1}) = F_{effect1} \tag{B.1}$$

$$F_{SP1} = F_{effect1} + F_{1\_mixer} \tag{B.2}$$

$$x_{SP2}(F_{SP2}) = F_{conc\_juice} \tag{B.3}$$

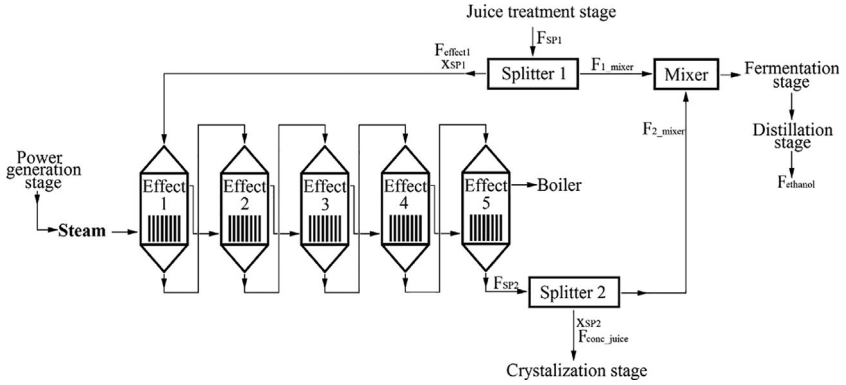

FIGURE B1 Concentration stage of the biorefinery

$$F_{SP2} = F_{conc\_juice} + F_{2\_mixer} \tag{B.4}$$

The biorefinery model does not comprise distillation columns or crystallization and crystals drying operations; the model has, instead, centrifuged wine and concentrated juice as output streams. In this way, in order to formulate the proposed optimization problem, ethanol production is translated into the ethanol mass flow rate in the centrifuged wine and sugar production is translated into the concentrated juice mass flow rate (F_{conc_juice} in Figure B1). In order to ensure that the section of cogeneration and biorefinery were energetically tied, an estimative of the energy consumed in the distillation and crystallization processes was adopted, based on Dias et al. [6] and was considered as equality constraint in the *Biorefinery model*, which includes all operations equations. In this way, it is possible to estimate the demand of electrical and thermal energy and, consequently, the surplus of electricity available for sale. In the cogeneration step (Figure B2), bagasse is sent to the boiler to be burned and to produce steam (10.0 MPa and 529.8°C) that feeds turbines and supplies steam to meet biorefinery requirements. Turbine 1 exhaust steam is at 2.1 MPa and 318.2°C, which is fed to Turbine 2, that reduces steam pressure to 0.25 MPa and steam temperature to 127.4°C. Finally, this steam is fed to Turbine 3, which operates between this pressure and 0.01 MPa. Turbine M receives a fraction of the outlet superheated steam (2.1 MPa and 318.2°C) (F_{outs_T1}) provided by Turbine 1 and feeds the biorefinery process with steam at 0.25 MPa and 176.5°C. Condensed steam, generated after passing through the turbines and the biorefinery process, is deaerated and returns to the boiler to be vaporized and reheated.

According to Dias et al. [6], to calculate the surplus of electrical power available for sale, the energy consumed by the distillation columns ($E_{\text{distillery}}$) and the minimum energy required for biorefinery operation ($E_{biorefinery} = 30$ MW) are deducted from the total electricity generated by the turbines ($E_{electrical\_total}$), presented in Equation B.5. Equation B.6 presents the function to estimate the energy consumed by the distillery that is function of F_{1_mixer} and F_{2_mixer} considering a factor of 0.0107 MW/tonne of ($F_{1\_mixer} + F_{2\_mixer}$). Finally, the electricity generated by the turbines ($E_{electrical\_total}$) is calculated by the sum of the four turbines, according to Equation B.7.

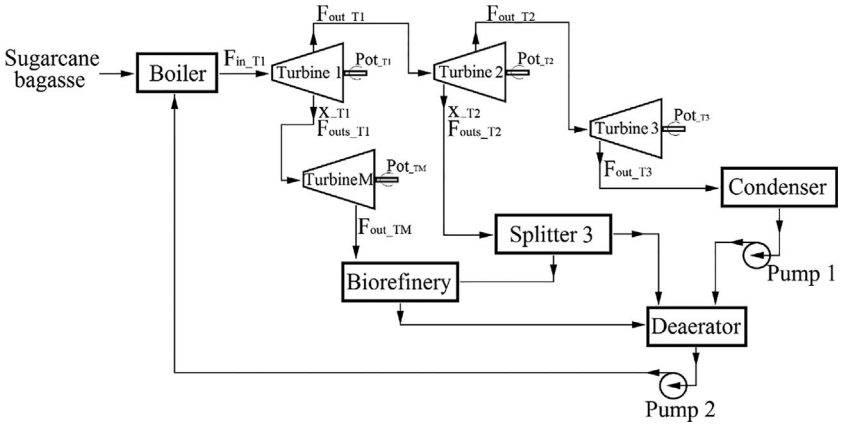

FIGURE B2 Flowsheet of cogeneration system of the biorefinery

$$E_{electrical} = E_{electrical\_total} - E_{distillery} - E_{biorefinery} \quad \text{(B.5)}$$

$$E_{distillery} = 0.0107(F_{1\_mixer} + F_{2\_mixer}) \quad \text{(B.6)}$$

$$E_{electrical\_total} = Pot_{\_T1} + Pot_{\_T2} + Pot_{\_T3} + Pot_{\_TM} \quad \text{(B.7)}$$

where, F_{1_mixer} is the flow rate of the juice outlet stream from splitter 1 that is diverted to fermentation step and F_{2_mixer} is the flow rate of the concentrated juice outlet stream from splitter 2 that is also diverted to fermentation step (see Figure B1).

Steam consumed by the biorefinery process must be provided by cogeneration system, according to Equations B.8 and B.9. This consumption is represented by the "Biorefinery" block in Figure B2 and this amount is represented in the model by Q_{steam}. Steam consumed in juice preparation ($Q_{\text{juice\_preparation}}$) is approximately 0.016 MWh/tonne of sugarcane. The steam consumed by the five effects evaporator is calculated based on its feed stream mass flow rate ($F_{effect1}$) [6].

$$Q_{steam} = Q_{\text{evaporators}} + Q_{\text{juice\_preparation}} \quad \text{(B.8)}$$

$$Q_{steam} = F_{\text{out\_TM}} + F_{\text{out\_T2}} \quad \text{(B.9)}$$